

1

AD-A257 429



TASK: UT20
CDRL: 04014
12 June 1992

UT20—PCTE Browser Tool Version Description Document Version 0.1

Informal Technical Data

DTIC
ELECTE
OCT 28 1992
S C D

424460
92-28328
85
PDS

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

STARS-TC-04014/002/00
12 June 1992

10 27 116

TASK: UT20
CDRL: 04014
12 June 1992

VERSION DESCRIPTION DOCUMENT
For The
SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS
(STARS)

PCTE Browser Tool
Version 0.1
SunOS Implementation

STARS-TC-04014/002/00
12 June 1992

Data Type: A005, Informal Technical Data

CONTRACT NO. F19628-88-D-0031
Delivery Order 0008

Prepared for:

Electronic Systems Division
Air Force Systems Command, USAF
Hanscom AFB, MA 01731-5000

Prepared by:

Paramax Systems Corporation
Tactical Systems
12010 Sunrise Valley Drive
Reston, VA 22091

DTIC QUALITY INSPECTED 2

Accession For	
THIS CDRL	<input checked="" type="checkbox"/>
REF ID	<input type="checkbox"/>
Unpublished	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Avail and/or	
Dist	Special
A-1	

TASK: UT20
CDRL: 04014
12 June 1992

VERSION DESCRIPTION DOCUMENT
PCTE Browser Tool
Version 0.1
SunOS Implementation

Principal Author(s):

Michael J. Horton, Paramaz, Valley Forge Labs

Date

Approvals:

Thomas E. Shields

6/17/92

Task Manager *Dr. Thomas E. Shields*

Date

(Signatures on File)

TASK: UT20
CDRL: 04014
12 June 1992

VERSION DESCRIPTION DOCUMENT

PCTE Browser Tool

Version 0.1

SunOS Implementation

Change Record:

<i>Data ID</i>	<i>Description of Change</i>	<i>Date</i>	<i>Approval</i>
STARS-TC-04014/002/00	Original Issue: Describes alpha release software, version 0.1. Implements a PCTE OMS browser, based on version 0.5 of the Reusable Graphical Browser component (RE: STARS-SC/03714/004/00).	12 June 1992	<i>on file</i>

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 12 June 1992	3. REPORT TYPE AND DATES COVERED Version Description Document		
4. TITLE AND SUBTITLE PCTE Browser Tool Version Description Document		5. FUNDING NUMBERS F19628-88-D-0031		
6. AUTHOR(S) Paramax System Corporation				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Paramax System Corporation 12010 Sunrise Valley Drive Reston, VA 22090		8. PERFORMING ORGANIZATION REPORT NUMBER STARS-TC-04014/002/001		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Department of the Air Force Headquarters Electronic Systems Division Hanscom AFB MA 01731-5000		10. SPONSORING / MONITORING AGENCY REPORT NUMBER 04014		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Distribution "A"		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) The PCTE Browser Tool (PBT) is designed to graphically display parts of a PCTE object base. Selected objects in the object base and the relationships amongst these objects are displayed at the PBT user's request. The PBT is intended to complement text-oriented commands such as <code>obj_list_links</code> and <code>obj_list_attr</code> that are included with the Emeraude PCTE 1.5 release—commands intended to be invoked from the text-oriented <code>esh</code> command shell. PBT version 0.1 is an alpha release of the browser.				
14. SUBJECT TERMS PCTE, Browser Tool		15. NUMBER OF PAGES 19		
		16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR	

Contents

1 SCOPE	1
1.1 Identification	1
1.2 System Overview	1
2 RELATED SOFTWARE	1
3 VERSION DESCRIPTION	2
3.1 Inventory of Contents	2
3.1.1 Subdirectory: pbt01/code	2
3.1.2 Subdirectory: pbt01/bin	2
3.1.3 Subdirectory: pbt01/X-Resources	2
3.1.3.1 Subdirectory: pbt01/X-Resources/PCTE-bitmaps . . .	2
3.2 Adaptation Data	3
3.2.1 Operating Environment	3
3.2.2 Development Environment	3
3.2.3 Configuration-Unique Data	3
3.3 Installation Instructions	4
3.3.1 Build Procedure	4
3.3.2 Executable Installation Procedure	6
3.3.3 Installing the X Resource Files	6
3.3.4 Locating the X Resource Files	7
3.4 Potential Problems	7
3.5 Enhancements	7
4 USER FEEDBACK	8
A Appendix: Inventory of Contents	9
B Appendix: Unix Installation Scripts	11
B.1 File: Build_PBT.var	11
B.2 Script: Build_PBT.csh	17

1 SCOPE

1.1 Identification

Version Description Document,
PCTE Browser Tool (PBT),
Version 0.1,
SunOS Implementation

1.2 System Overview

The PCTE Browser Tool (PBT) is designed to graphically display parts of a PCTE object base. Selected objects in the object base and the relationships amongst these objects are displayed at the PBT user's request. The PBT is intended to complement text-oriented commands such as `obj_list_links` and `obj_list_attr` that are included with the Emeraude PCTE 1.5 release—commands intended to be invoked from the text-oriented `esh` command shell. PBT version 0.1 is an alpha release of the browser.

2 RELATED SOFTWARE

The PBT is an instance of the Reusable Graphical Browser (RGB), a generic graphical browser for the display of networks of nodes and arcs. In the case of the PBT, the nodes displayed by the RGB are PCTE objects, and the arcs are PCTE links. PBT version 0.1 was developed using RGB version 0.5.

The PBT is an X Window System application, and requires the installation of X11. It has been built and tested using Release 4 of X11; however, it is expected that it also should be usable under X11R3 or X11R5.

It was developed using the Paramax STARS Ada implementation of Ada/Xt and Ada implementation of some MIT Athena and Hewlett Packard widgets, version 3.3.

The PBT is ultimately intended for use in an ECMA PCTE environment, and has been implemented using the ECMA-162 Ada programming bindings to PCTE. However, in the absence of a conforming ECMA PCTE implementation, the PBT has been built on top of the Emeraude V12.2 PCTE implementation, using the subset implementation of the ECMA Ada binding developed by Paramax STARS (version 0.1).

3 VERSION DESCRIPTION

3.1 Inventory of Contents

The PBT distribution is structured as shown below. The top-level directory **pbt** includes PostScript (**VDDpbt.ps**) and clear ASCII text (**VDDpbt.tty**) versions of this document. It contains a complete directory listing of the PBT distribution (**Contents.tty**, reproduced herein as **Appendix A**). It also contains a PostScript version of the PBT user manual (**USERpbt.ps**). Finally, it contains the following subdirectories, described below:

- pbt01/code**
- pbt01/bin**
- pbt01/X-Resources**
- pbt01/X-Resources/PCTE-bitmaps**

3.1.1 Subdirectory: **pbt01/code**

This directory contains the Ada source code for the PBT. It also contains the C shell scripts and associated support files needed to rebuild the PBT.

3.1.2 Subdirectory: **pbt01/bin**

This directory contains the Sun-4 executable for the browser, built using the SunAda 1.0 Ada compiler. This is the directory into which the build process moves the PBT executable after a successful compile and link.

3.1.3 Subdirectory: **pbt01/X-Resources**

This directory contains the **PCTE-Browser.black_n.white** and **PCTE-Browser.color** files describing the X resource values used by the browser for black-and-white and color monitors, respectively. These values specify such characteristics of the PBT as the dimensions to be used for the various windows created by the browser. This directory also contains the **PCTE-bitmaps** subdirectory, described below.

3.1.3.1 Subdirectory: **pbt01/X-Resources/PCTE-bitmaps**

This directory contains X bit maps for the icons used by the PBT. These icons represent the different types of nodes (e.g., File) and relationships (e.g., Composition and Reference links) recognized by the browser.

3.2 Adaptation Data

3.2.1 Operating Environment

Sun-4 Workstations with at least 32 megabytes of main memory

SunOS, Version 4.1.2

X Window System, Version 11, Release 4

Use of any "standard" X window manager (e.g., TWM or MWM)

Emeraude PCTE V12.2

Note that this release of the PBT has not been tested either Release 3 or 5 of X Window System, Version 11; however, it is expected that the PBT would be operational under either of these two other releases of X11.

3.2.2 Development Environment

Sun-4 Workstation with 32 megabytes of main memory

SunOS, Version 4.1.2

Ada/Xt Toolkit, Version 3.3

Reusable Graphical Browser, Version 0.5

X Window System, Version 11, Release 4

SunAda version 1.0 Ada compilation system

Emeraude PCTE V12.2

ECMA PCTE Ada Bindings, Version 0.1

3.2.3 Configuration-Unique Data

There is only one explicit dependency in the PBT itself to UNIX, in its use of the "exit" procedure as part of the PBT termination processing. (This procedure is accessed via the Ada *pragma INTERFACE* capability in the code file *utilities_b.a.*) However, there are more such dependencies on UNIX in the Ada/Xt implementation. Refer to the VDD for Ada/Xt version 3.3 for more information.

The PBT makes extensive use of ECMA PCTE Ada bindings, which, in its current implementation is highly dependent upon the Emeraude V12.2 PCTE implementation.

3.3 Installation Instructions

The sections below describe the steps needed to:

- build the PBT executable
- install the PBT executable in the environment
- install the PBT's X Resource file, **PCTE-Browser**

(See the accompanying PBT user manual for details on how to use the browser.)

3.3.1 Build Procedure

This section describes the procedure for compiling and linking the PBT program using the SunAda 1.0 Ada compilation system from Sun Microsystems.

Before proceeding with the build of the PBT, first verify that the following assumptions are correct:

- The entire PBT delivery contents have previously been loaded onto the local file system. For purposes of these installation instructions, the top-level directory for the PBT delivery shall be referred to as */local/pbt01*.
- Ada/Xt, version 3.3, has previously been loaded onto the local file system, at a location to be referred to below as */local/adart33*.
- The Ada/Xt libraries for **Xlib**, **Xt**, **Widgets** and **C** have previously been built using the SunAda 1.0 Ada compiler. See the VDD for the Ada/Xt release for information on how to build these libraries. They are assumed to be found in the following UNIX directories:
 - */local/adart33/Build_SunAda1.0/Xlib*
 - */local/adart33/Build_SunAda1.0/Xt*
 - */local/adart33/Build_SunAda1.0/Widgets*
 - */local/adart33/Build_SunAda1.0/C*
- The Reusable Graphical Browser, version 0.5, has previously been loaded onto the local file system, at a location to be referred to below as */local/rgb05*.
- The RGB library has previously been built using the SunAda 1.0 Ada compiler. See the VDD for the RGB release for information on how to build this library. This library is assumed to be found in the following UNIX directory:
 - */local/rgb05/Build_SunAda1.0/rgb*

- The Emeraude PCTE implementation, version V12.2, has been loaded onto the local file system, at a location to be referred to below as */local/pcte12.2*.
- The ECMA PCTE Ada Bindings implementation version 0.1, has been loaded onto the local file system, at a location to be referred to below as */local/adapcte01*.
- The ECMA PCTE Ada Bindings has previously been built using the SunAda 1.0 Ada compilation system. See the VDD for the ECMA PCTE Ada Bindings release for information on how to build this library. This library is assumed to be found in the following UNIX directory:

– */local/adapcte01/Build_SunAda1.0*

- The Xlib archive file corresponding to the X11R4 delivery has previously been created. Consult with your local system administrator for the exact location of the Xlib archive file on your system. For purposes of this discussion, it is assumed that this file can be found at:

– */usr/lib/X11/libX11.a*

To build the PBT, first edit the `code/Build_PBT.var` file to reflect the actual operating environment. This file (listed in its entirety in Appendix B.1) initializes the environment variables used by the rest of the build process. Variables that must be initialized include the following:

- PBT – the top level directory of the PBT distribution
- RGB – the directory containing the RGB Ada library built using SunAda 1.0
- AdaXt – the top level directory of the Ada/Xt implementation's build directories
- LIBX – the pathname of the X11R3 or X11R4 Xlib archive
- COMPILERPATH – the pathname of the top-level directory of the SunAda 1.0 compilation system
- PCTE – the directory containing the SunAda library for the ECMA PCTE Ada bindings
- PCTE_ROOT – the top level directory of Emeraude's PCTE delivery

Once the `code/Build_PBT.var` file has been edited, the rest of the compiling and linking of the PBT is fully automated. Simply *cd* to the PBT distribution's `code` directory and execute the `code/Build_PBT.csh` C shell script (shown in its entirety in Appendix B.2), as in the following example:

```
% cd /local/pbt/code
% Build_PBT.csh >& Build.out &
```

This script creates a new directory called `Build_SunAda1.0` below the top-level PBT directory in which the actual build will take place. That is, the PBT's SunAda Ada library will be created in and the link will take place in this new directory.

Assuming that the build is successful, the executable PBT will be moved by the build script into the `bin` directory beneath the top-level PBT directory—replacing any version of PBT previously in that directory.

Note that all of the environment variables that are set within `code/Build_PBT.var` are set *conditionally*, i.e., only if these variables have not been set *outside* of the build process (e.g., within the user's `.login` file). This means that the person invoking the build process can set these values prior to invoking `code/Build_PBT.csh` script—without explicitly editing `code/Build_PBT.var`.

3.3.2 Executable Installation Procedure

Assuming that the build is successful, the executable PBT will be moved into the `bin` directory beneath the top-level PBT directory—replacing any version of PBT previously in that directory.

The PBT executable could be installed as a static context within the PCTE object base prior to its first use. However, it can also be accessed from within PCTE by placing it in a UNIX directory that is part of the UNIX `PATH` environment variable. Therefore, it is assumed that the user will either include the PBT's `bin` directory in the user's path, or will copy the PBT executable to another directory already in the path (e.g., `/usr/local/bin`).

3.3.3 Installing the X Resource Files

A number of UNIX files associated with the PBT must be on-line at the time that the PBT is executed:

- A set of files describing the bitmaps to be used for the various node and link icons.
- The "X resource file" associated with the PBT, `PCTE-Browser`, describing such information as which bitmap to use for which type of object, what dimensions to use for the various widgets used by the PBT, etc.

In the case of the `PCTE-Browser` file, two different versions are supplied in the release:

- `PCTE-Browser.color` - for use on color monitors
- `PCTE-Browser.black_n_white` - for use on monochrome monitors

Each of these two versions has a line in it identifying the directory containing the bitmaps. If the PBT has been installed on the local system at any location other than `/local/pbt01`, then the following line within these two versions of the PCTE-Browser file must be modified to reflect the actual location of the installed bitmaps:

```
*bitmapFilePath:  /local/pbt01/X-Resources/PCTE-bitmaps
```

3.3.4 Locating the X Resource Files

Prior to executing the PBT, the specific version of the PCTE-Browser file appropriate to the type of monitor to be used for the PBT session must be copied to—or linked within—a UNIX directory under the name PCTE-Browser. In addition, the PBT user must identify this directory to the PBT by making it the value of the XAPPLRESDIR environment variable.

3.4 Potential Problems

1. The PCTE-Browser is required to be in the directory identified by the XAPPLRESDIR environment variable. If it is not found in this directory, or if the XAPPLRESDIR variable is not properly set, then the PBT will terminate almost immediately with the following error message:

```
Error in kernel:: exception_handler: unexpected SIGILL code 16
```

2. The PBT sometimes has problems when destroying (quitting) View windows. The expected PBT behavior is for an Alert Box to pop up when a problem has been detected. However, occasionally, the PBT will get itself into an infinite loop, appearing to be locked up, i.e., not responding to any mouse or keyboard events. In this case, the PBT session must be terminated from outside of the browser. If the PBT was started in the foreground, this can be done simply by hitting `cntrl-C` from the `xterm` window from which the PBT was started. If the PBT was started in the background, then the UNIX `kill` command must be used.

3.5 Enhancements

Possible future enhancements to the PBT include:

- Improved graph layout algorithms.
- Migration to a conforming ECMA PCTE environment.
- Replacement of the STARS Ada/Xt implementation by a commercial Ada binding to Motif widgets.

4 USER FEEDBACK

This version of PBT is considered an "alpha" release. The primary purpose of the release is to encourage experimentation with the software and to solicit feedback from the PCTE community to assist us in improving the product. Thus, we would greatly appreciate your comments, suggestions, and criticisms.

A Appendix: Inventory of Contents

NOTE: "*" identifies executables; "/" identifies directories.

pbt01:

Contents.tty

USERpbt.ps

VDDpbt.ps

VDDpbt.tty

X-Resources/

bin/

code/

doc/

pbt01/X-Resources:

PCTE-Browser.black_n_white

PCTE-Browser.color

PCTE-bitmaps/

pbt01/X-Resources/PCTE-bitmaps:

c_rel.xbm

f_node.xbm

i_rel.xbm

o_node.xbm

p_node.xbm

p_rel.xbm

r_rel.xbm

s_rel.xbm

pbt01/bin:

PBT*

pbt01/code:

Build_PBT.csh*

Build_PBT.var

browser_instance.a

browser_params.a

browser_params_b.a

callbacks.a

callbacks_b.a

globals.a

main.a

pcte_layout.a

pcte_layout_b.a

pcte_object_create.a

12 June 1992

STARS-TC-04014/002/00

pcte_support.a
pcte_support_b.a
pcte_text_io.a
pcte_text_io_b.a
pipe_int.c
static_cmds.a
static_cmds_b.a
static_menus.a
static_menus_b.a
utilities.a
utilities_b.a

B Appendix: Unix Installation Scripts

B.1 File: Build_PBT.var

```
1 #
2 # Edit these lines and leave them uncommented if you do not want to
3 # be prompted for the environment variables
4 #
5 setenv PBT          /local/pbt01
6 setenv RGB          /local/rgb05/Build_SunAda1.0/rgb
7 setenv AdaXt        /local/adaxt33/Build_SunAda1.0
8 setenv LIBX          /usr/lib/libX11.a
9 setenv COMPILERPATH /local/SunAda
10 setenv PCTE          /local/pcteAda01/Build_SunAda1.0
11 setenv PCTE_ROOT     /local/pcte12.2
12
13 # Variables that need not be modified:
14 setenv OS            4.1
15 setenv Sun           4
16 setenv COMPILERNAME  sunada
17 setenv COMPVERSION   SunAda1.0
18 setenv TARGET        $PBT/Build_$COMPVERSION
19
20 #
21 # Define the location of the PBT source code directories.
22 #
23 if ( ! $?PBT ) then
24     echo ""
25     echo "Specify path to top level PBT directory "
26     echo "(e.g. /local/pbt01) "
27     echo ""
28     echo -n "          PBT = "
29     setenv PBT $<
30     echo ""
31 endif
32 if ( ! -e $PBT ) then
33     echo ""
34     echo "** $PBT does not exist **"
35     echo "** Script aborted **"
36     echo ""
37     unsetenv PBT
38     exit -1
39 endif
40
41 #
```

```
42 # Define the location of the RGB source code directories.
43 #
44
45 if ( ! $?RGB ) then
46     echo ""
47     echo "Specify path to directory containing RGB Ada library "
48     echo "(e.g. /local/rgb05/Build_SunAda1.0/rgb) "
49     echo ""
50     echo -n "          RGB = "
51     setenv RGB $<
52     echo ""
53 endif
54 if ( ! -e $RGB ) then
55     echo ""
56     echo "** $RGB does not exist **"
57     echo "** Script aborted **"
58     echo ""
59     unsetenv RGB
60     exit -1
61 endif
62
63
64 #
65 # Define the location of the dependencies.
66 #
67 if ( ! $?AdaXt ) then
68     echo ""
69     echo "Specify path to top level AdaXt build directory "
70     echo "(e.g. /local/adaxt33/Build_SunAda1.0) "
71     echo ""
72     echo -n "          AdaXt = "
73     setenv AdaXt $<
74     echo ""
75 endif
76 if ( ! -e $AdaXt ) then
77     echo ""
78     echo "** $AdaXt does not exist **"
79     echo "** Script aborted **"
80     echo ""
81     unsetenv AdaXt
82     exit -1
83 endif
84
85 setenv WIDGETS      $AdaXt/Widgets # Sample Widgets Ada libraries
86 setenv XLIB         $AdaXt/Xlib    # Ada/Xlib bindings Ada libraries
```

```
87  setenv XT          $AdaIt/Xt      # Ada/Xt Toolkit Ada libraries
88  setenv XMU         $AdaIt/Xmu     # Ada/X Miscellaneous Utilities Ada libraries
89
90  #
91  # Define the location of the X11R3/R4 Xlib archives
92  #   where XLIB = path to the X11 Xlib object archive (e.g./usr/lib/libX11.a)
93  #
94  if ( ! $?LIBX ) then
95      echo ""
96      echo "Specify the path to the X11 Xlib object archive "
97      echo "(e.g. /usr/lib/libX11.a ) "
98      echo ""
99      echo -n "          LIBX = "
100     setenv LIBX $<
101     echo ""
102 endif
103 if ( ! -e $LIBX ) then
104     echo ""
105     echo "*** $LIBX does not exist ***"
106     echo "*** Script aborted ***"
107     echo ""
108     unsetenv LIBX
109     exit -1
110 endif
111
112 #
113 # Define C Language compilation variable
114 #
115 setenv CC          " cc -g -c "
116
117 #
118 # Determine the Ada compilation system to use
119 #
120 #
121 # Establish a path to the SunAda compilation system
122 #
123 if ( ! $?COMPILERNAME || ! $?COMPVERSION || ! $?COMPILERPATH ) then
124     echo ""
125     echo "Please select your compiler name: [sunada] "
126     echo ""
127     echo -n " COMPILERNAME = "
128     setenv COMPILERNAME $<
129     echo ""
130     switch ($COMPILERNAME)
131         case SunAda:
```

```
132 case SUNADA:
133 case sunada:
134     echo -n "Are you building with SunAda1.0? [y,n](n) "
135     set COMVERSION = $<
136     echo ""
137     switch ($COMVERSION)
138     case Y:
139     case y:
140         set COMVERSION = SunAda1.0
141         breaksw
142     case N:
143     case n:
144     default:
145         set COMVERSION = SunAda
146         echo ""
147         echo "Warning! Software has only been tested using SunAda 1.0."
148         breaksw
149     endsw
150 breaksw
151 default:
152     echo ""
153     echo "You must specify a compiler name."
154     echo ""
155     unsetenv COMVERSION
156     exit -1
157     breaksw
158 endsw
159
160 echo ""
161 echo "Specify path to the compiler (e.g. /local/SunAda)"
162 echo ""
163 echo -n "    COMPILERPATH = "
164 setenv COMPILERPATH $<
165 if ( ( $COMPILERPATH == ) || ( ! -e $COMPILERPATH/bin/ada ) ) then
166     echo ""
167     echo "** Cannot find Ada compiler in $COMPILERPATH/bin **"
168     echo "** Script aborted **"
169     echo ""
170     unsetenv COMPILERPATH
171     exit -1
172 endif
173 endif
174 if ( -e $COMPILERPATH/bin/ada ) then
175     if ( $COMPILERNAME == "sunada" ) then
176         setenv COMPILERBIN $COMPILERPATH/bin
```

```
177     setenv COMPILE      "$COMPILERBIN/ada -v -00 "
178     setenv LINK         "$COMPILERBIN/a.ld "
179 endif
180 else
181     echo ""
182     echo "*** Cannot find $COMPILERPATH/bin/ada ***"
183     echo "*** Script aborted ***"
184     echo ""
185     unsetenv COMPILERPATH
186     exit -1
187 endif
188
189
190 #
191 # Define the Destination of the PBT build
192 #   where TARGET = path to build destination (e.g. $PBT/Build_SunAda1.0)
193 #
194 if ( ! $?TARGET ) then
195     echo ""
196     echo "Specify the path to the TARGET directory "
197     echo "(Defaults to $PBT/Build_${COMPVERSION}) "
198     echo ""
199     echo -n "          TARGET = "
200     setenv TEMP $<
201     echo ""
202     if ( $TEMP == ) then    # check for null entry
203         setenv TARGET $PBT/Build_${COMPVERSION}
204         unsetenv TEMP
205     else
206         setenv TARGET $TEMP
207         unsetenv TEMP
208     endif
209 endif
210
211 echo ""
212 echo "          TARGET = $TARGET"
213 echo ""
214 echo "          RGB = $RGB"
215 echo "          PBT = $PBT"
216 echo "          AdaIt = $AdaIt"
217 echo "          XLIB = $XLIB"
218 echo "          IT = $IT"
219 echo "          IMU = $IMU"
220 echo "          WIDGETS = $WIDGETS"
221 echo "          PCTE_ROOT = $PCTE_ROOT"
```

```
222 echo ""
223 echo "          LIBX = $LIBX"
224 echo "  COMPILERNAME = $COMPILERNAME"
225 echo "    COMPVERSION = $COMPVERSION"
226 echo "  COMPILERPATH = $COMPILERPATH"
227 echo "        COMPILE = $COMPILE"
228 echo "          LINK = $LINK"
229 echo "            OS = $OS"
230 echo "          Sun = $Sun"
231
232 #
233 # Create the directories for the build
234 #
235 if ( ! -d $TARGET ) mkdir $TARGET
236 if ( ! -d $TARGET/rgb ) mkdir $TARGET/rgb
237 if ( ! -d $TARGET/application ) mkdir $TARGET/application
```

B.2 Script: Build_PBT.csh

```
1  #! /bin/csh -f
2  echo ""
3  echo "Defining installation-dependent variables"
4  echo ""
5  source Build_PBT.var
6
7
8  if ! -e $TARGET mkdir $TARGET
9
10 cd $TARGET
11
12 echo ""
13 echo "Building Ada libraries for the PCTE Browser Tool (PBT)"
14 echo "-- a sample application of the Reusable Graphical Browser --"
15 echo ""
16 if ( $COMPILERNAME == "sunada" ) then
17     if (! -e $RGB/ada.lib) then
18         echo "Sorry. RGB must be built first. Script aborted."
19         exit -1
20     endif
21
22     if (-e ada.lib ) a.rmlib -f          # clean out old library
23
24     $COMPILERBIN/a.mklib -f $TARGET $COMPILERPATH/verdirxlib
25
26     echo ""
27     echo "Establishing dependencies"
28     echo ""
29
30     $COMPILERBIN/a.path -i $PCTE
31     $COMPILERBIN/a.path -i $RGB
32     $COMPILERBIN/a.path -i $WIDGETS
33     $COMPILERBIN/a.path -i $XNU
34     $COMPILERBIN/a.path -i $XT
35     $COMPILERBIN/a.path -i $XLIB
36 else
37     echo "Sorry. Only SunAda is currently supported. Script aborted."
38     exit -1
39 endif
40
41 echo ""
42 echo "Building TARGET directory with symbolic links to source code"
43 echo ""
```

```
44  foreach file ($PBT/code/*.a $PBT/code/*.c)
45    if ( ! -e ${file:t} ) ln -s $file ${file:t}
46  end
47
48  cd $TARGET
49
50  echo ""
51  echo "Compiling the C source"
52  echo ""
53  cc -c pipe_int.c
54  if ( $status != 0 ) exit $status
55
56  echo ""
57  echo "Compiling the PBT source"
58  echo ""
59  $COMPILE browser_params.a
60  if ( $status != 0 ) exit $status
61  $COMPILE browser_params_b.a
62  if ( $status != 0 ) exit $status
63  $COMPILE browser_instance.a
64  if ( $status != 0 ) exit $status
65  $COMPILE pcte_object_create.a
66  if ( $status != 0 ) exit $status
67  $COMPILE pcte_support.a
68  if ( $status != 0 ) exit $status
69  $COMPILE pcte_support_b.a
70  if ( $status != 0 ) exit $status
71  $COMPILE pcte_text_io.a
72  if ( $status != 0 ) exit $status
73  $COMPILE pcte_text_io_b.a
74  if ( $status != 0 ) exit $status
75  $COMPILE pcte_layout.a
76  if ( $status != 0 ) exit $status
77  $COMPILE pcte_layout_b.a
78  if ( $status != 0 ) exit $status
79  $COMPILE globals.a
80  if ( $status != 0 ) exit $status
81  $COMPILE static_menus.a
82  if ( $status != 0 ) exit $status
83  $COMPILE static_cmds.a
84  if ( $status != 0 ) exit $status
85  $COMPILE callbacks.a
86  if ( $status != 0 ) exit $status
87  $COMPILE static_menus_b.a
88  if ( $status != 0 ) exit $status
```



```
89 $COMPILE static_cmds_b.a
90 if ( $status != 0 ) exit $status
91 $COMPILE utilities.a
92 if ( $status != 0 ) exit $status
93 $COMPILE utilities_b.a
94 if ( $status != 0 ) exit $status
95 $COMPILE callbacks_b.a
96 if ( $status != 0 ) exit $status
97 $COMPILE main.a
98 if ( $status != 0 ) exit $status
99
100 echo ""
101 echo "Linking the objects"
102 echo ""
103 set objects = ($RGB/call_ada.o $PCTE/util.o pipe_int.o)
104 set libs = ($AdaXt/C/lib.a $LIBX $PCTE_ROOT/lib/libemer.a)
105 $LINK -v main $objects -o PBT $libs
106 if ( $status != 0 ) exit $status
107 mv PBT $PBT/bin/PBT
108
109 echo ""
110 echo "Build Complete"
```